LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A276 361

DTIC ACCESSION NUMBER

LEVEL

(1)

INVENTORY

WL-TR-94-1007

DOCUMENT IDENTIFICATION

Feb 94

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I ☑
DTIC TRAC ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DTIC
S ELECTE D
MAR 1 1994
C

DATE ACCESSIONED

H A N D L E W I T H C A R E

DATE RETURNED

94 2 24 157

DATE RECEIVED IN DTIC

13P6 94-06194

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM JUN 90 70A   DOCUMENT PROCESSING SHEET   PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

WL-TR-94-1007

TESTING OPERATIONAL FLIGHT PROGRAMS (OFPS)

CHARLES P. SATTERTHWAITE

FEBRUARY 1994

FINAL REPORT FOR 05/10/93-05/13/93

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

This technical report has been reviewed and is approved for publication.

Charles P. Satterthwaite
Project Engineer
WL/AAAF-3

[illegible] L. Harris
Chief, WL/AAAF-3

Donna M. Morris
Chief, WL/AAAF

FEB 1994 FINAL 05/10/93--05/13/93

TESTING OPERATIONAL FLIGHT PROGRAMS (OFPS)

C
PE
PR 9994
TA 00
WU 00

CHARLES P. SATTERTHWAITE

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

WL-TR-94-1007

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

10

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED UL

TESTING OPERATIONAL FLIGHT PROGRAMS (OFPs)

Charles P. Satterthwaite
United States Air Force
Wright Laboratory
Avionics Logistics Branch, WL/AAAF Bldg 635

1. SUMMARY

The ability to accurately test a system which you are developing is a highly desirable feature in the engineering design process. The ability to model your system's environment and to exercise your system, in that environment, is also highly desirable.

Operational Flight Programs are the software programs of avionics embedded computer systems. Not only is it desirable to be able to test and model Operational Flight Programs, it is essential. The consequences of not performing accurate Operational Flight Program testing can be devastating. Some of these include premature weapon releases, erroneous flight instrument displays, and complete system failure.

In order to test Operational Flight Programs, there are several things one must know about the Operational Flight Program, its weapon system host, its support environment, and how to generate and perform its test. This paper will address these issues as it develops a strategy to test an Operational Flight Program.

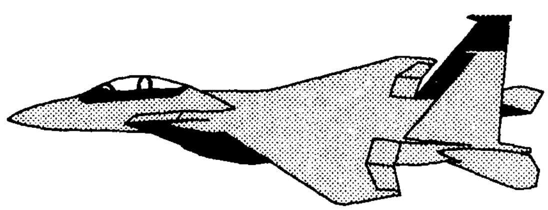

Figure 1

2. INTRODUCTION

Embedded computers are increasingly called upon to provide high-tech solutions to complex multiple threat environments for today's generation of weapon systems (Ref 6). The empowering of an embedded computer is its software, which is the Operational Flight Program.

In understanding the role of an OFP, one must thoroughly understand the threat, the weapon system, the mission, the embedded computer system, and the complex testing issues associated with OFPs (Ref 7).

The ultimate success of an updated Operational Flight Program is that the new OFP becomes an operational version. Although several layers of testing must be successfully passed before OFPs are operationally acceptable. Flight tests are expensive, as are full-up simulations. But some confidence can be gained through evaluating the OFP through a simulation environment. The simulation environment takes advantage of real-time avionics hardware, realistic simulation software, and the adaptability of advanced technologies to provide a capability for testing the weapon system, the weapon system's subsystems and units, and the weapon system's software (the OFPs) (Refs 7,9).

Testing Operational Flight Programs requires an understanding of: how OFP architecture and processes work; how an OFP is changed; the major components of an OFP and its support environment; the OFP's interaction with its users/maintainers; OFP testing/validation issues; breadth and depth of OFP tests; and how OFP test results are analyzed and interpreted (Refs 4,7).

3. OFP ARCHITECTURES AND FUNCTIONS

The Operational Flight Program literally is the software portion of a embedded computer system. The computer and its peripheral interfaces make up the system hardware. The hardware enabled by the OFP software describes the whole system.

The OFP is made up of a series of modules which represent the functions of the weapon system. These functions describe the mission phases which the weapon system can perform. Mission phases include preflight, takeoff/time to cruise, outbound cruise, SAM (surface to air missile) evasion, descent, penetration, bomb delivery, climb, air-to-air combat, inbound cruise, loiter, and approach and landing. Function types include communication (external/internal), IFF (identification friend or foe), navigation, guidance, steering, control, target acquisition/identification, stores management, weapon delivery, and threat warning. The modules of the OFP include executive, control and display, air-to-air, air-to-ground, navigation, communication, heads up display, vertical situation display, gun, missiles, overload warning, and visual identification. A module type, such as controls and displays, might contain multiple modules which are prioritized according to the timing requirements of the functional calls of the OFP. The OFP is required to process real time interrupt driven schedules, which are handled by the executive modules. The modules of the OFP are made up of machine level object code. Access to this object code by OFP maintainers is through a higher order language source code which can be compiled to the object code. Examples of higher order languages used in maintaining OFPs are Ada, COBOL, and FORTRAN (Refs 2,6,7,8).

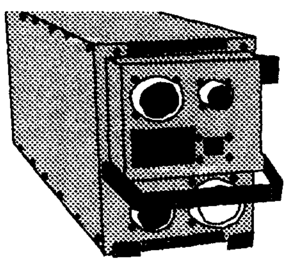

Figure 2

The embedded computer system (see Figure 2) has partitioned memory which is filled with some type of machine level object (binary) code. The OFP is loaded into this partitioned memory, and when enabled, empowers the whole system to perform its desired functions. Each embedded computer system has an instruction set which is burned into its Read Only Memory (ROM). The instruction set allows the embedded computer maintainer access to the OFP as well as the capability to optimize the remaining partitioned memory. The level of sophistication of a embedded computer system is a function of the programming expertise of its OFP maintainers, its instruction set, its memory, its hardware, and its throughput (Ref 7).

4. HOW IS AN OFP CHANGED?

Given a working OFP in a working system, why would changes ever be necessary? One reason is that the users of the system require an altered mission. As an example, a pilot would request a clearer display under some dynamic threat condition. Another reason to change OFPs is that some flaw is discovered while the embedded computer system is operational. Some combination of events might cause partial or total system failure, prompting a review and redesign in the effected areas of hardware, software, or both.

Given the task of changing an OFP (making a new version or even a new block cycle), several steps are followed to bring about the change. First, the requested change(s) is diagnosed so that it is clearly understood. Once the OFP maintainer thoroughly understands the change request, an analysis is made of the OFP areas which need to be altered. Usually the OFP is made up of a series of modules with specialized functions. A typical change might impact three modules of an OFP which contains 40 modules. The OFP maintainer will next isolate these modules by making copies of them and implementing design changes to the copies. The OFP maintainer integrates these modules by linking them together with the other unaltered modules to form a unique OFP. The OFP maintainer's final task is to thoroughly test this modified OFP by putting it through an acceptance test procedure. For a sizable OFP with several changes, a number of OFP maintainers would follow these procedures simultaneously, and then a lead OFP maintainer would integrate and test the new OFP (Ref 7).

5. MAJOR COMPONENTS OF OFP TESTING AND DEVELOPMENT

5.1 The Target Processor

In order to perform various levels of testing on OFPs, the OFPs embedded computer (also called the target processor) must be available and accessible. The actual target processor (see Figure 2) is often used by OFP maintainers to build a mockup support environment by which they can access and test their OFP changes. When these target processors are used, an environment has to be available which stimulates the processor input requirements and receives the processor output. Some examples of inputs are power, cooling, and peripheral interfaces (such as pilot commands and avionics suite inputs). Examples of outputs include pilot displays, as well as, command and control logic for other processors (Ref 7).

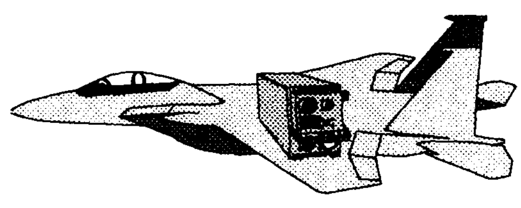

Figure 3

5.2 The Support Environment

In order to maintain an OFP, the maintainers require a dedicated computer system and a simulation environment.

The dedicated computer system (see Figures 4 and 5) allows the maintainer to access the OFP's object code as well as to copy and alter this code. The simulation environment allows maintainers to run the OFPs which enables them to interactively debug and test.

The dedicated computer system provides system conventions which are configuration management, security procedures, and proper operation of the dedicated computer system.

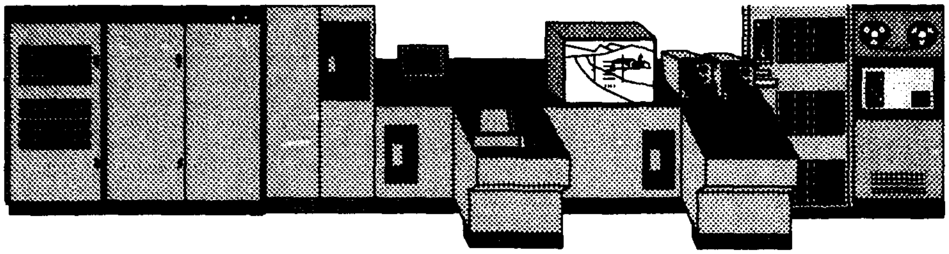

Figure 4

The hardware of a dedicated computer system usually includes mainframe computers (or powerful engineering workstations), various types of printers, disk storage devices, networking, and several access terminals.

Embedded computers and dedicated computers are frequently confused as being the same. These are actually quite different. The embedded computer is the target processor which is part of the weapon system. The dedicated computer is outside of the weapon system and is used to support the software run on the embedded computer system.

5.3 Simulation Environment

OFPs must have a means by which to operate in real-time, that is, loading them up in their target processor and exposing them to the range of conditions (or a reasonable subset of those conditions) encountered while operational. This allows the maintainer to actively debug the OFP. The degree of complexity of the OFP's environment is directly related to the complexity of this simulation environment. In the case of a typical fire control computer, a method to represent the full-up avionics suite and the dynamic environment of the fighter is required. An interface to all cockpit controls and switches, as well as, an interface between the dedicated computer system and the simulation environment is necessary. Finally, competent maintainers,

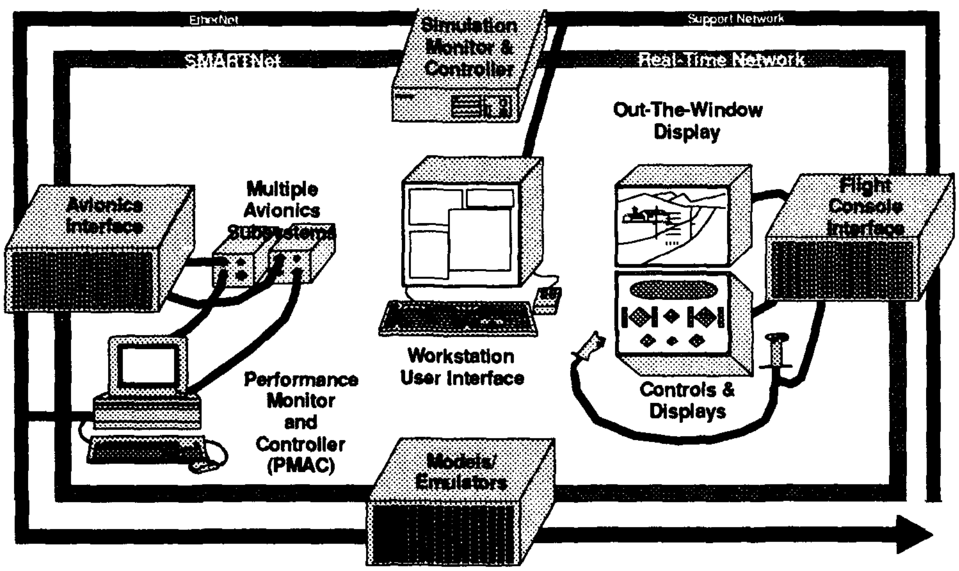

Figure 5

who know how to make the system work, are essential.

The simulation can range from a fully operational weapon system (flight testing is very expensive) to an all-software engineering workstation. Usually the simulation is a representative set of the weapon system's LRUs (Line Replaceable Units) with software emulating the cockpit and the dynamic environment.

Interaction with the simulation environment is through the dedicated computer system. Simulation utilities hosted on the dedicated computer system allow the loading of an OFP into its target processor and also allow the OFP to be exercised dynamically or statically. These utilities also allow recording, patching, debugging, freezing , and the initialization of the OFP (Refs 2,6,7,8).

5.4 The Avionics Integrated Support Facility (AISF)

The facility which houses the dedicated computer system(s) and the simulation environment(s) is the Avionics Integrated Support Facility (AISF). Another name for the AISF is the Centralized Software Support Activity (CSSA). The AISF supports one or more embedded computer systems and the associated OFPs.

6. OFP TESTING ISSUES

6.1 The Requirement to Test

The requirement to test is related to the confidence desired of the targeted system or subsystem. Low level testing might be sufficient for minor operational adjustments such as flight-line data entry. But processes affecting life support, terrain following radar, and navigation, to name a few, require highly integrated testing. These processes often require specialized testing which depend on critical resources such as specialized hardware, test equipment, test software patches, and OFP maintainer expertise (Refs 1,2,4,5,6,7,8,9).

6.2 What Is An Acceptable Level Of Testing?

This question is best asked of the crew members of the OFP's weapon system since it is their task to complete missions, as well as, survive. The quality and quantity of OFP testing affects their lives. At a minimum, crew members must be assured of the normal operating conditions of their weapon system. Additionally, maximum performance capabilities should be made available, as well as, a fail safe capability (Refs 2,8).

6.3 Iterative Nature Of OFP Testing

Usually OFPs are not acceptable in their first cut, even when they go through Operational Test and Evaluation. Five or six cycles through the testing process is not unusual. Much of this is related to the complex nature of OFPs, poor interpretation of OFP engineering change requests, and changing mission requirements midstream in OFP development (Ref 6).

7. TYPES OF OFP TESTS

7.1 The Acceptance Test Procedure

The OFP maintainers primary test is the acceptance test procedure (ATP). This test is designed to check out an OFP to a degree that it can be released with confidence to flight test and then operational test and evaluation.

The ATP is a chronological check of the OFP's responses to inputs. Inputs include switch positioning, preset conditions such as altitude or airspeed, and hardware interrupts to name a few. The OFP is loaded into its embedded computer, hosted on its simulation environment, and required to respond to these inputs in the form of static or dynamic displays, which can be checked against expected results.

The ATP for a typical fire control computer could contain 200 or more independent tests of varying degrees of complexity. The reliance of an OFP acceptance test procedure to be visually verified and to be manually performed requires several weeks to complete (Ref 7).

7.2 The Baseline Acceptance Test Procedure

The baseline acceptance test procedure (ATP) is the ATP which complemented the most recent version of the OFP (the last block cycle change). An ATP should be developed concurrently with its OFP. That is to say, any additions, deletions, or modifications to the OFP should be paralleled by the ATP (Ref 7).

7.3 Unit Tests

A unit test is the lowest level of testing. With respect to an OFP, a unit test is at the module level. As an example, there might exist some type of looping mechanism within a module. The check of this loop might be with a clock to time the loop or a count down mechanism to track the number of loop iterations (Refs 7,9).

7.4 Subsystem Tests

A subsystem test combines units to represent a functional set of an OFP. In typical fire-control computers, these subsystems might include the set of air-to air modules or the set of control and display modules. Checks for these types of subsystems include setting a value in one module, running the OFP, and inspecting values in other modules against expected values (Refs 2,7,9).

7.5 Integrated Tests

Integrated testing, as seen in Figure 6, can represent several layers of OFP testing. Integrated testing includes the exercising of an OFP's complete module set. It is here that the subsystems are checked out against each other and against the OFP's target processor environment. The integrated test can become increasingly complex as the environment is more dynamically modeled. An example of an increasingly dynamic environment is changing from

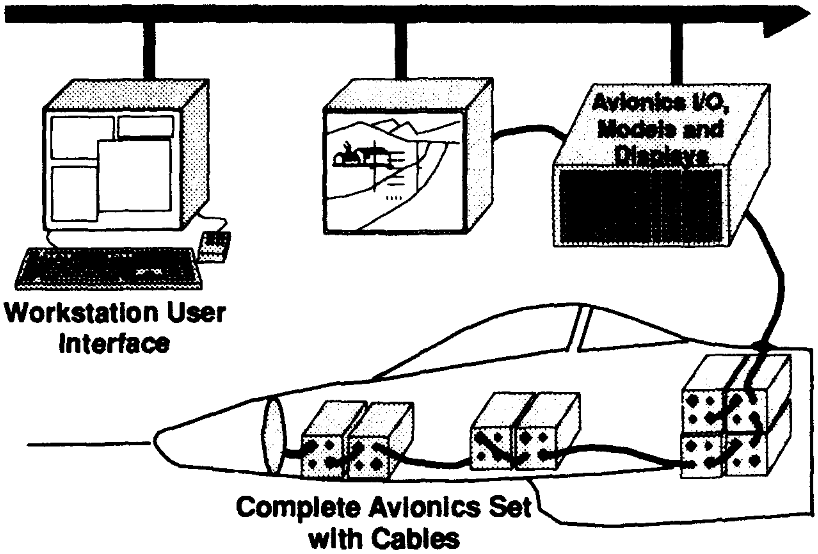

Figure 6

modeled radar inputs to actual radar inputs being driven by a separate radar OFP (Refs 2,3,6,7,8).

7.6 Static Tests

Static tests are tests which are not time dependent. Given an input, or a combination of inputs, there should be an expected response. As an example, in gun mode, a gun reticle should appear on the pilot displays. The gun reticle is a circle displayed to a pilot on a Heads Up Display (HUD) and a Visual Situation Display (VSD). The static test is that when the gun mode is initiated, the gun reticle is or is not present. If it is not present, it has failed the test.

7.7 Dynamic Tests

Dynamic tests are much more complicated than static tests, since they are time dependent. They might require a sequence of inputs over some time interval in order to ensure proper functioning of the OFP. An example of a dynamic test is to observe an expected Signal-to-Noise Ratio (SNR) improvement, as range decreases on a target being tracked with radar. The difficulty of this test is that it requires an OFP maintainer who can visually verify the test case. The maintainer has to know from experience what a sequence of responses should indicate. The quality of OFP testing in the dynamic cases is often limited to the experience level of OFP maintainers available for testing (Refs 2,7).

7.8 Classified Tests

Arrangements must be made for classified testing of OFPs. This requires the facilities and maintainers to be cleared to the level of the classification of testing. It also requires a means of properly storing and maintaining classified testing documentation. It is often convenient to isolate classified portions of OFP testing, so that non-classified OFP testing can be accomplished with minimal restrictions (Ref 7).

7.9 Automated Tests

As the complexity of OFPs increases, the ability to manually perform acceptance test procedures (ATPs) decreases. Also, the ability to fully and accurately test OFPs decreases. One successful method to increase the OFP maintainer's ability to test OFPs is to utilize automated techniques. For example, if in the process of manually

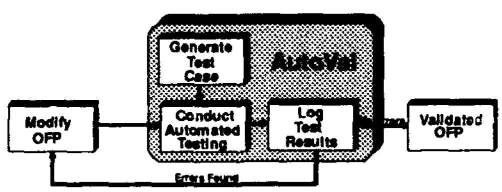

Figure 7

running an ATP test case, a sequence of switch and dial position can be captured throughspecial test software, then that portion of the ATP test case can be automated. Using techniques like this should reduce errors and the time to run through ATP and free OFP maintainers to develop more comprehensive test cases (Refs 2,3,4,5,6,7,8,9).

7.10 Operational Test And Evaluation

Operational test and evaluation is where the OFP must meet the approval of those who will use it. These users have their own check-out procedures which can include live firing of munitions, lock-on and destruction of drones, navigational exercises, to mention a few. Operational test and evaluation is the final test of a complete weapon system being fully integrated together. Several different OFPs can be evaluated during operational test and evaluation. Operational test and evaluation often finds system and subsystem errors, which were undetectable in the OFP maintainer's simulation environment. Often, OFP version updates are refined by OFP maintainers through the information they receive from Operational test and evaluation (Refs 7,9).

7.11 Developmental Test And Evaluation

Sometimes during the block cycle or version update of OFPs, a more dynamic environment than the Avionics Integrated Support environment is required. Some test situations can only be examined through the actual exercising of the complete system in its real environment. Developmental test and evaluation provides OFP maintainers with this option, usually through the provision of instrumented flight test aircraft. These instrumented aircraft can accommodate specific tests in an actual operational environment. An example of this is the recording of narrow band and wide band data in an air-to-air engagement scenario which can be analyzed for specific OFP performance parameters (Refs 7,9).

8. HOW IS AN OFP TESTED?

8.1 What is Normal?

Before originating or extending the OFP's acceptance test procedure, a baseline must be established which outlines the system's normal performance parameters and the environmental conditions which the system will experience. This baseline will influence the testing design decisions throughout the weapon system's lifecycle. In this baseline, design considerations must include the weapon system's embedded computer systems, their OFPs, and their interaction.

Performance parameters include all of the avionics of the system such as altitude, air speed, angle of attack, directional indication, and engine thrust. Performance is also the ability of the air crew to interact with the system through controls and displays. Performance also includes the system's interaction with its environmental conditions through the use of its communications, navigation, radar, electronic warfare suite, and weapons. Consideration should be given to the performance of the system's OFPs. Are the OFPs operating optimally? Are there unused resources that can be better shared? Are there potential bottlenecks or failures that can be avoided?

Environmental conditions are those situations that the weapon system will be exposed to. In the normal course of a mission, what does the weapon system experience? The weapon system is prepared for its mission at its home base. It leaves its home base enroute to its mission, it is refueled enroute, it maneuvers to avoid threats enroute, it performs its mission, and it reverses its enroute to return to home base. Several environmental conditions have been identified in this mission scenario. First, a maintenance or mission preparation environment is identified. Second, a navigational environment is pointed out. Third, a friendly air-to-air refueling environment is called for. Fourth, a threat environment is shown. Fifth, the mission performance environment occurs. And finally, there is the return environment.

In each of the above environments (plus several others), every possibility of weapon system configuration must be identified. The OFP's influence on every weapon system configuration, and subsystem configuration, in every environment in response to the weapon system's performance parameters gives the foundational basis for the OFP acceptance test procedure. The baseline OFP acceptance test takes into account every parameter, every environmental situation, and any combination of parameters and environmental situations to generate test cases which exercise these various situations (Refs 1,2,3,6,7,8,9).

8.2 What could Impact Normality?

Given a comprehensive understanding of the system's performance and the various environments in which it can be exercised, what changes, threats, or failures should be anticipated?

One of the greatest benefits of using embedded computers and software in weapon systems is that these systems can be reconfigured and adapted to changing mission requirements and evolving threats more readily than older hardware intensive systems. There is a cost associated with this benefit. In a highly integrated weapon system, small changes can effect large testing areas. It is important to know, before changes are made, how these changes influence the entire system, and what changes in testing need to be made to facilitate them.

The threat environment is constantly changing. It is necessary for weapon systems to be carefully tuned to certain threats in order to defeat or avoid them. What happens when a unique unanticipated threat is put up against the weapon system? If possible, unique threats should be anticipated and planned for in testing scenarios. Evolving and break-through technologies often translate into new threats. By keeping pace with these new technologies, potential threats can be included in the test plans.

System and subsystem failure should also be considered when anticipating potential impacts on normal testing. At what degraded capability could the system operate if various subsystems were disabled (Refs 1,2,3,6,7,8,9).

8.3 Generation of the An Acceptance Test Procedure?
Having established normal and abnormal performance criteria of the system, a comprehensive acceptance test procedure can be established. This test would begin by identifying and describing every possible configuration of the weapon system against every possible environment that the system would encounter. This test would then identify, describe, and anticipate every abnormal situation which could impact the system and its subsystems. With the inventory of configurations derived, a set of test cases would then be generated to exercise these configurations. The actual utilization of these test cases would determine the requirements for each test case such as the static or dynamic testing, degree of integration with other OFP components, simulation resources, and the number of iterations of the test case. The compilation of all this information is the acceptance test procedure.It should be noted that using present techniques to complete an acceptance test procedure, as described for a modern weapon system, would take several man months, with many of the configurations untestable (Refs 1,2,3,6,7,8,9).

8.4 Passing The Test?
What qualifies an OFP as passing its acceptance test procedure? The obvious answer is, you pass when you sufficiently satisfy every test case in your acceptance test procedure.

Because the maintenance of OFPs has not been prioritized in the procurement process, what is used for an acceptance test procedure is greatly stripped down from what has been suggested. Current OFP acceptance test procedures are heavily dependent on the OFP maintainer's subjective experience. The passing or failing of an OFP acceptance test procedure is based on how these OFP maintainers feel about their weapon system. Though not scientific or repeatable, this has been sufficient to field reliable systems.

Future OFP acceptance test procedures will demand identifiable and repeatable processes in order to guarantee weapon system reliability. The increase in configurational situations alone will disqualify the subjective expert method of passing OFP acceptance test procedures. Future OFP acceptance test procedures will require a comprehensive identification, description, and anticipation of the situations the system will and might experience. In addition, future OFP acceptance test procedures will need methods to test these situations.

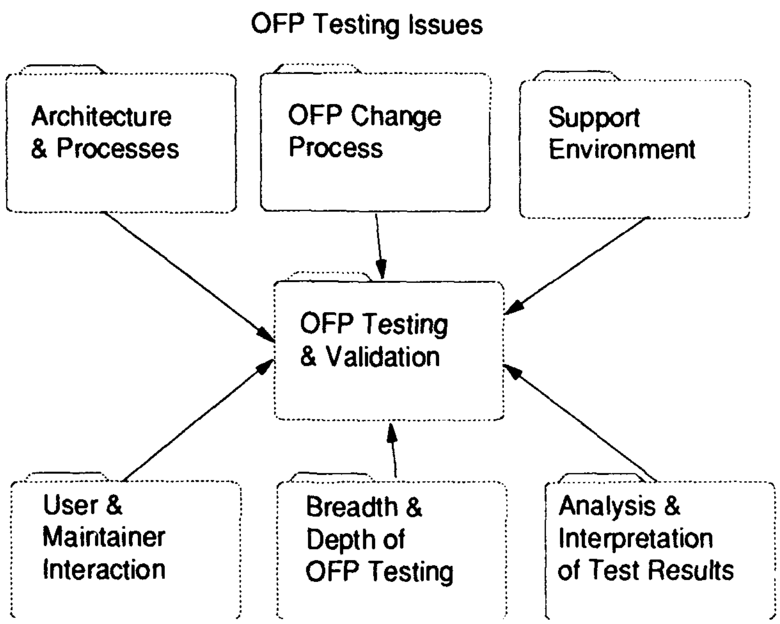

Figure 7

8.5 Increasing Levels of Integration
The nature of weapons platforms is to increase in complexity. The ability to increase in complexity has been largely facilitated by using embedded computer systems and software. These embedded systems are increasingly linked together (or integrated) to take advantage of shared resources. The consequences of increased integration is increased complexity in the ability to test the weapon

system. When subsystems are isolated, changes in those subsystems have little or no impact on the overall weapon system. When these subsystems are integrated through some shared resources, changes in a subsystem potentially impacts all of its sharing partners.

Unfortunately, as systems have become more complex, the capability to test these systems has not kept pace. This is largely due to the fact that the procurement process has not provided for or anticipated the maintenance requirements of advanced avionics software. It is well documented that 70% or more of a system's life cycle cost will be in the maintenance of that systems software. A large portion of this cost lies in the system's testing (Refs 7,9).

For every increased level of system integration, at least equal thought, design, and resources should be dedicated to testing. This will require new analysis, methodologies, and testing techniques (Refs 2,3,4,5,6,7,8).

8.6 Need for Advanced Technologies

In order to assure the successful operation of current and future avionics weapon systems, as well as, the growing number of system platforms implementing highly integrated embedded computer systems and software, advanced avionics testing technologies must be encouraged and accelerated. Some of the areas to be pursued include: improved instrumentation techniques; development of integrated diagnostics techniques (especially in the area ofsoftware integrated diagnostics); continued emphasis on automated testing techniques; development of advanced verification and validation techniques; expansion of avionics software reuse libraries; improved simulation and testing environments; increased implementation of hypermedia and virtual reality technologies into the OFP testing environments; and continued development of human factor engineering (Refs 2,3,4,6,7,8).

The encouragement and implementation of these types of technologies will: enable the weapon system to monitor itself while it is operational; return from its mission and give its maintenance staff a comprehensive performance and diagnostics report; suggest new techniques for evaluating complicated highly integrated OFPs, and identify reserve capabilities and opportunities for the weapon system (Refs 2,8).

9. CONCLUSIONS

Operational Flight Programs hosted on embedded computer systems have greatly extended the capabilities of avionics weapon systems. These extensions have increased the: weapon systems lethality; the air crews survivability; and the capability of the system to be reconfigured as well as decreased the weapon system turn around time In order to be further extended, a new emphasis must be placed on the testing of Operational Flight Programs. This new emphasis is dependent on the inclusion of advanced avionics technologies into existing and planned Avionics Integrated Support Facilities. It is also dependent on all individuals involved in the acquisition and maintenance of weapon systems containing OFPs to be aware of what it takes to have confidence in the software.

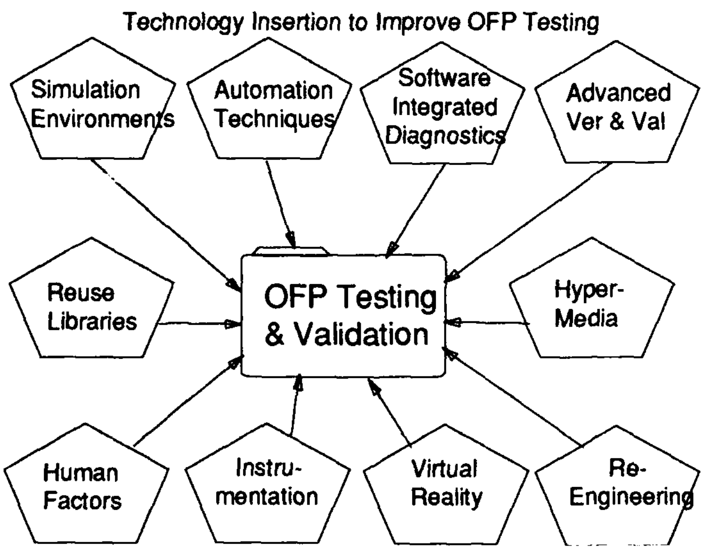

Figure 8

10. REFERENCES

[1] DOD-STD-2167A, Military Standard Defense System Software Development, 29 February 1988.

[2] Blais,R.R.,Neese,R.E.,Noharty,,K.L., "Modular Embedded Computer Software (MECS) Final Report, Volume II", Air Force Technical Report WL-TR-92-1101, Wright-Patterson AFB, OH, June 1992.

[3] Harris,R.L., "Laboratory Concepts in Avionics Software", IEEE/AIAA/NASA 9th Digital Avionics Systems Conference Proceedings, 15-18 October 1990, Virginia Beach VA.

[4] Harris,R.L.,Jackson,O., "Software Engineering Tools For Avionics Embedded Computer Resources", IEEE/AIAA National Aerospace Electronics Conference Proceedings, 21-25 May 1990, Dayton OH.

[5] Jones,J.V., *Engineering Design Reliability, Maintainability, Testability*, TAB Books Inc., Blue Ridge Summit, PA, 1988.

[6] Morris,D.M., "Avionics Operational Flight Program Software Supportability", IEEE/AIAA/NASA 9th Digital Avionics Systems Conference Proceedings, 15-18 October 1990, Virginia Beach VA.

[7] Satterthwaite,C.P., *Maintaining An Operational Flight Program (OFP)*, Air Force Technical Memorandum WL- TM-91-123, Wright-Patterson AFB, OH, 1991.

[8} Satterthwaite,C.P.,"Getting a Handle On Designing For Avionics Software Supportability And Maintainability",IEEE/AIAAA National Aerospace Electronics Conference Proceedings,18-22 May 1992,Dayton OH

[9] Computer Resource Acquisition Course, "Course Materials Volume I and II", System Acquisition School, 6575TH School Squadron, Air Force Systems Command, Brooks AFB, TX, December 1991.